A Unified Model for Collaborative Sentiment Classification

Author Name[1*]
Author Name,
Author Name
[1]address
email address

Abstract

Sentiment classification is a fundamental in the areas of natural language processing and data mining. However, most of existing works perform sentiment classification based on local text, ignoring other texts that have similar sentiment labels. In this paper, we proposed a novel collaborative sentiment classification model (CSC). CSC is inspired by the collaborative filtering technique that users with similar rating behaviours are supposed to have similar ratings in the future. In CSC, we speculate that users with similar rating behaviours are more likely to write reviews of similar sentiment toward a product. In other words, CSC bases on user similarities to select similar documents for collaborative sentiment classification. To this end, we decompose CSC into three parts, namely user-product interaction (UPI) component, document encoding (DE) component and speculative similar document (SSD) component. The UPI component models user-product interactions, and encodes user/product ratings behaviours into user/product embeddings. The DE component utilizes learned user/product embeddings to capture the informative word vectors for comprising document representations. The SSD component aggregates documents written by similar user toward a given product for collaborative sentiment classification. Since the user similarities are calculated based on user embeddings that encode user rating behaviours, the aggregated documents are more likely to have similar sentiments.

The three components are seamlessly integrated into a unified model, and one advantage of the proposed model is that these three components are jointly optimized, and they can mutually complement each other to enhance the analysis of specific document sentiment. We conduct extensive experiments on two public datasets, and demonstrates the advantage of the proposed model over the state-of-the-art baselines.

Introduction

Sentiment classification is to classify the sentiment or polarity of a given text, and it plays an essential role in natural language processing and data mining. Sentiment classification has a wide application on online review websites due to their rapid growth and huge amount of review data for classification. Sentiment classification problem can be approached at different level of granularities (e.g. aspect, sentence, and document), and we focus on document-level sentiment classification that aims to determine the overall sentient of a user review about a product.

Various methods have been proposed to tackled sentiment classification, from feature-engineering-based methods (Taboada et al. 2011; Ming et al. 2014) to deep-learning-based methods (Tai, Socher, and Manning 2015; Yang et al. 2016; Li et al. 2019b). Recently, many works(Amplayo et al. 2018; Tang, Qin, and Liu 2015; Chen et al. 2016a) propose to incorporate user and product information into sentiment classification. For example, Chen et al. (2016a) utilize user product embeddings to calculate attention weights over the word vectors, and represent document as the weighted sum of the word vectors. However, most of these methods mainly exploit local text information while ignoring sentiment consistency (Dou 2017) among the documents of the same user/product. Notice that even if Amplayo et al. (2018) explore similar users/products to address cold-start problem, the similar users/products are mainly used to calculate attention weights over the word vectors, hence it is still limited by the information in a single text.

UPDMN (Dou 2017) is the state-of-the-art model that takes advantage of the sentiment consistency for sentiment classification. The rationale underlying UPDMN is that documents written by (about) a user (product) are more likely to have the same sentiment distribution. Therefore, it stores documents of each user/product in a memory, and utilizes an attention mechanism to aggregate those documents to produce a document representation for classification. Even though UPDMN has shown promising results, it suffers from the following two problems:(a) for classifying a document d, it mainly uses the weighted aggregation of the document representations in the memory while ignoring the representation of d. (b) documents of a user/product do not necessarily have the same sentiment, simply aggregating those documents can introduce noises and negatively affect model performance.

To this end, we propose a novel collaborative sentiment classification model (CSC). CSC is inspired by the basis

[*]Primarily Mike Hamilton of the Live Oakfile Press, LLC, with help from the AAAI Publications Committee

idea of collaborative filtering that users with similar rating behaviours in the past are more likely to have similar ratings in the future (He et al. 2017). CSC is based on the speculation that users with similar historical rating behaviours are more likely to write documents of similar sentiments toward a product. In other words, CSC leverages user similarities to select informative documents of the products for collaborative sentiment classification. Those selected documents are termed as speculative similar documents (SSDs) in this paper. To accomplish this, we first model **user-product interactions** to collaboratively factorize user/product rating behaviours into low dimensional embeddings. The user embeddings reflect user rating behaviours, and can be used to find similar users. In the meanwhile, we encode the documents into high-level representations. To capture informative word vectors to represent documents, we incorporate user product embeddings into the **document encoding** process with an attention mechanism. Finally, given a document d written about a product and all the other documents about the product, we speculatively select the informative documents based on the similarities between user of d and that of the other documents. Then we aggregate the representations of those **speculative similar documents** to collaboratively determine the sentiment of d.

The modeling of user-product interactions, document encoding and speculative similar documents are then integrated into a unified model. Even though the modeling of user-product interactions and document encodings have been sufficiently studied in previous works (Chen et al. 2016a; He et al. 2017), however, most of them mainly study one of these two sub-tasks alone. In this work, we propose to combine the aforementioned three components into a unified model. One advantage of the unified model is that those three components can be mutually reinforce each other to enhance the sentiment classification. For example, the modeling of user-product interactions can learn task-specific user product embeddings, and it guides the document encoding to capture the most informative word vectors for comprising document representations. In addition, the SSDs are selected based on the similarities between user rating behaviours, which are encoded in user embeddings learned from user-product interactions.

To conclude, the contributions of this work are as follows:

- We propose a novel method to select informative documents as auxiliary information for collaborative sentiment classification. The method is based on the speculation that users with similar rating behaviours are more likely to generate documents of similar sentiment toward a product.
- We seamlessly incorporate user-product interactions modeling, document encoding and SSDs modeling into a unified model. The three component are interdependent, and can be complementary to each other for sentiment classification.
- We demonstrate that the proposed model can outperform state-of-the-art models with publicly available datasets, and validate the effectiveness of each component of the proposed model.

The Proposed Model

Problem Definition

The data records in this work are presented as tuples: $t_k =< u_i, p_j, d_k >$, where $d_k \in D$ is the k-th review (document), and $u_i \in U$ and $p_j \in P$ is the corresponding user and product. The tuple indicates that the user u_i has written a review d_k about the product p_j, and the task is to predict the sentiment distributions or ratings for the tuple t_k. Notice that each document is a sequence of words, $d_k = \{w_1^k, \cdots, w_{l_k}^k\}$, where w_t^k is the t-th word in d_k and l_k is the length of d_k. For simplicity, we denote $D(p_j)$ as all the documents written about product p_j, and $U(p_j)$ the corresponding users that write the documents, namely $D(p_j) = \{d_p | d_p \text{ is written about } p_j\}$ and $U(p_j) = \{u_p | u_p \text{ writes } d_p \in D(p_j) \text{ about } p_j\}$.

Architecture Overview

The overall framework of CSC is presented in Fig.1. CSC can be decomposed into three components, namely user-product interaction component (UPI), document encoding component (DE) and speculative similar document component (SSD). For a given tuple $< u_i, p_j, d_k >$, the UPI component takes u_i, p_j as input, and outputs a user-product interaction vector $\mathbf{z}_{ij}$. Specifically, we map the user and product into low dimensional embeddings, and apply multi-layer perceptrons on these embeddings for capturing their interactions. The user product embeddings are randomly initialized and need to be learned in the training process. The input of the DE component are the words in d_k, and the output is a document vector $\mathbf{d}_k$. In this paper, we map the words into word embeddings, and employ stacked 1-dimension CNNs to extract high-level word vectors from those embeddings, and then incorporate user product embeddings (i.e. $\mathbf{u}_i, \mathbf{p}_j$) to capture the most informative word vectors for comprising the document vector $\mathbf{d}_k$. In SSD component, we select similar users for u_i from $U(p_j)$, and aggregate the corresponding documents in $D(p_j)$ into a SSD vector $\mathbf{d}_{ssd}$. Since the similarities between u_i and users in $U(p_j)$ are calculated based on their embeddings which encode the user rating behaviours, the selected users are suppose to have similar rating behaviours as u_i, and the corresponding documents are more likely to have similar sentiment as d_k.

User-Product Interaction

The UPI component is based on the latent factor model that decomposes a rating matrix into low dimensional user and product embeddings. The interactions between user-product embeddings can capture user preferences over the products, hence they can be used for predicting sentiment ratings. The rationale underlying UPI modeling is that users with similar historical behaviours are supposed to have similar ratings in the future.

Inspired by previous works (He et al. 2017), we employ multiple neural networks as the basic model for capturing deep user-item interactions. We denote $\mathbf{u}_i, \mathbf{p}_j$ as the embeddings of user u_i and product p_j respectively. These embeddings can be indexed from randomly initialized embedding matrices and need to be learned. Given the embeddings, the

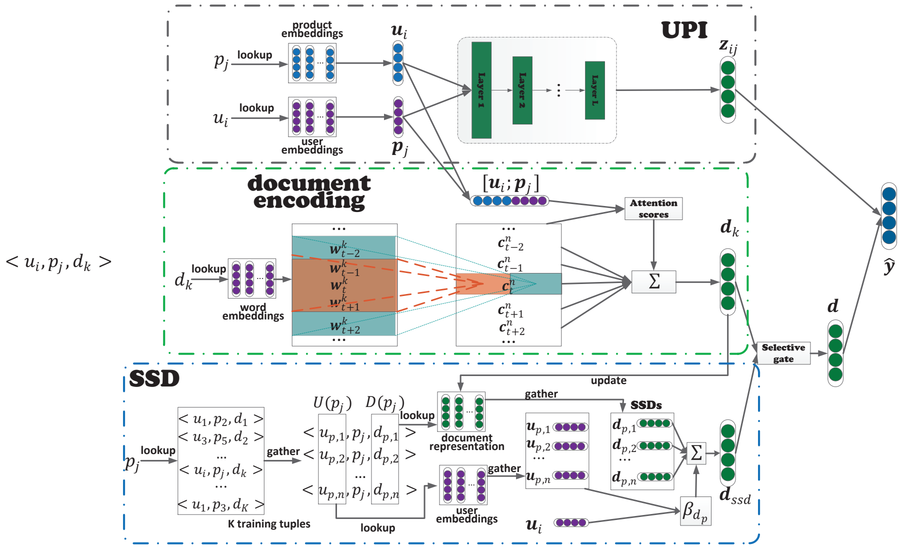


Figure 1: Overall architecture of CSC.

sentiment class distribution of u_i over p_j can be estimated as:

$$
\begin{aligned}
\mathbf{z}_0 &= [\mathbf{u}_i; \mathbf{p}_j; \mathbf{u}_i \circ \mathbf{p}_j] \\
\mathbf{z}_{ij} &= \phi_L(...\phi_1(\mathbf{z}_0)...) \\
\phi_l(\mathbf{z}_{l-1}) &= \sigma_l(\mathbf{W}_l \mathbf{z}_{l-1} + \mathbf{b}_l), l \in [1, L]
\end{aligned}
\tag{1}
$$

where $\circ$ is the element-wise multiply operation. L is the number of hidden layers in the neural network, and $\mathbf{W}_l, \mathbf{b}_l, \sigma_l$ are the weight matrix, bias vector and non-linear activation function of the l-th layer, respectively. $\mathbf{z}_{ij}$ can be regarded as the high-level UPI representation. Notice that the UPI modeling learns task-specific user product embeddings, and those embeddings are involved in computing attention scores over the word to obtain better document representations.

Document Encoding

The DE component takes a document d_k as input and outputs a distributed document representation $\mathbf{d}_k$. Each word w_t^k is mapped into a low dimensional embedding $\mathbf{w}_t^k \in \mathbb{R}^{d\times 1}$ through a embedding matrix using its index in the vocabulary. The word embeddings are then input to stacked 1-dimension CNNs to extract high-level word vectors. CNNs are proven powerful for sentiment classification (Kim 2014; McCann et al. 2017), as they are able to capture local semantics of various granularities when filter maps of different size are applied. Specifically, we apply a filter map $\mathbf{W}_f \in \mathbb{R}^{s\times d}$ of size s over a windows of s words center at word w_t^k, and produce a new single feature $c_{t,f}$:

$$
c_{t,f} = g([\mathbf{w}^k_{t-(s-1)/2}; \cdots; \mathbf{w}^k_{t+(s-1)/2}]^T * \mathbf{W}_f + \mathbf{b}_f) \tag{2}
$$

where $[;]$ is the concatenation operation, g is a non-linear function, $\mathbf{b}_f \in \mathbb{R}^{k_f}$ is a bias vector and $*$ is the convolutional operation. In practice, we use filter maps of different sizes (i.e. $\mathbf{W}_{f_1}, \mathbf{W}_{f_2}, \cdots$) to convolves over document d_k, and concat the corresponding features into a vector at each position of the document:

$$
\mathbf{c}_t = [c_{t,f_1}; \cdots; c_{t,f_i}; \cdots; c_{t,f_n}] \tag{3}
$$

where $\mathbf{c}_t \in \mathbb{R}^n$ can be viewed the word vector extracted from CNNs for word w_t^k, and c_{t,f_i} is the single feature produced by filter map $\mathbf{W}_{f_i}$ center at w_t^k. Notice that $\mathbf{c}_t$ has dimensionality equals to the number of filter maps. The produced feature vectors $\{\mathbf{c}_t\}_{t=1}^{l_k}$ are then input to multi-layer CNNs for extraction high-level word vectors. We denote the word vectors after n-layer CNNs as $\{\mathbf{c}_t^n\}_{t=1}^{l}$.

Words in the documents are noisy, and not all words are equally informative for the documents, as users may have different preferences over the words for describing different products. Therefore, we employ an attention mechanism to capture the informative word vectors, and aggregate them into document representations:

$$
\mathbf{d}_k = \sum_{t=1}^{l_k} \alpha_t \mathbf{c}_t^n \tag{4}
$$

where $\mathbf{d}_k$ can be viewed as the representation of document d_k, and α_t is the attention score assigned to the t-th word in d_k. The attentions scores $\{\alpha_t\}_{t=1}^{l_k}$ need to be specialized by the corresponding word, user and product. Therefore, The attention score of $\mathbf{c}_t^n$ can be obtained as follows,

$$
\begin{aligned}
\beta_t &= \mathbf{v}^T tanh(\mathbf{W}_c \mathbf{c}_t^n + \mathbf{W}_{up}[\mathbf{u}_i; \mathbf{p}_j] + \mathbf{b}_d) \\
\alpha_t &= \frac{exp(\beta_t)}{\sum_{t=1}^{l_k} exp(\beta_t)}
\end{aligned}
\tag{5}
$$

where the matrices $\mathbf{W}_c, \mathbf{W}_{up}$ and the vectors $\mathbf{b}_d$, $\mathbf{v}$ are model parameters. β_t denotes the relevance score of $\mathbf{c}_t^n$ in representing document d_k and α_t is the softmax normalization of β_t.

Speculative Similar Document

Existing works (Zheng, Noroozi, and Yu 2017) demonstrate that all documents of a user/product can be exploited to alleviate data sparsity problem and boost application specific performance (e.g. recommendation), as documents about a user/product reveal the attributes about the user/product from different points of view. However, in sentiment classification, incorporating reviews of different sentiment ratings has negative impact on the classification performance. To address this issue, the proposed SSD component aggregate those documents written by similar users, as documents written by users with similar rating behaviour are more likely to have similar sentiments toward a product.

Without loss of generality, given a tuple $t_k = < u_i, p_j, d_k >$, we aim to select documents in $D(p_j)$ based on the similarities between the rating behaviours of u_i and that of users in $U(p_j)$. Those selected documents are speculated to have similar sentiment distributions, and are aggregated to obtain a SSD representation:

$$\mathbf{d}_{ssd} = \sum_{d_p \in D(p_j)} \beta_{d_p} \mathbf{d}_p \tag{6}$$

where $\mathbf{d}_{ssd}$ is the SSD representation of d_k, $\mathbf{d}_p$ is the document representation of $d_p \in D(p_j)$ obtained with multi-layer CNNs as described in Eq.(4), and β_{d_p} is the corresponding aggregation weight of d_p. The aggregation weights are parameterized by the similarity between u_i and $u_p \in U(p_j)$:

$$\begin{aligned} s(u_i, u_p) &= \mathbf{v}_{ssd}^T tanh(\mathbf{W}_u \mathbf{u}_i + \mathbf{W}_p \mathbf{u}_p + \mathbf{b}_s) \\ \beta_{d_p} &= \frac{I(u_i, u_p) exp(s(u_i, u_p))}{\sum_{u_p \in U(p_j)} I(u_i, u_p) \cdot exp(s(u_i, u_p))} \\ s.t.\ u_p &\in U(p_j) \end{aligned} \tag{7}$$

where $\mathbf{u}_p$ is the embedding of u_p, the matrices $\mathbf{W}_u, \mathbf{W}_p$ and the vectors $\mathbf{b}_s$, $\mathbf{v}_{ssd}$ are model parameters. $s(u_i, u_p)$ measures the similarity between u_i and u_p, and $I(u_i, u_p)$ is an indication function that is used to select similar users. In this work, we adopt the thresholding mechanism (Wang et al. 2017; 2016) to define $I(u_i, u_p)$, which equals 1 if the similarity between the two users exceeds a certain threshold θ_s and 0 otherwise.

$$I(u_i, u_p) = \begin{cases} 1 & s(u_i, u_p) > \theta_s \\ 0 & s(u_i, u_p) < \theta_s \end{cases} \tag{8}$$

Notice that θ_s is not a constant in this work, but rather is left for the proposed model to learn (next section). Therefore, the document d_p is not selected for aggregating u_i's SSD representation, if the similarity between its user and u_i is below θ_s. The underlying rationale is straightforward, not all the documents written about a product have the same sentiment, and simply include all the documents for calculating the SSD representation can inevitably introduce noises.

Collaborative Sentiment Classification

Given the representation of a document d_k, $\mathbf{d}_k$, and its SSD representation $\mathbf{d}_{ssd}$, we propose a dual selective gate network to select between $\mathbf{d}_k$ and $\mathbf{d}_{ssd}$ into a pooled document vector, $\mathbf{d}$.

$$\begin{aligned} &\mathbf{g} = \sigma(\mathbf{W}_g[\mathbf{d}_k; \mathbf{d}_{ssd}] + \mathbf{b}_g) \\ &f(\theta_s) = \sigma((t_s - \theta_s)(\theta_s - t_d)) - 0.5 \\ &\mathbf{d} = (1 - f(\theta_s)) * \mathbf{g} \circ \mathbf{d}_k + f(\theta_s) * (1 - \mathbf{g}) \circ \mathbf{d}_{ssd} \end{aligned} \tag{9}$$

where $\circ$ is the element-wise multiplication, and $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function to limit the elements of its output in the range of [0,1]. t_s, t_d are the averages of the similarity scores of similar users (i.e. users with similarity score exceeds θ_s) and dissimilar users (i.e. users with similarity score smaller than θ_s) respectively. $f(\theta_s)$ denotes the degree of separation between similar users and dissimilar users. Specifically, if t_s and t_d are close to θ_s, then θ_s will be close to 0, indicating small differences between similar users and dissimilar users. In this case, there are high uncertainties in selecting the similar users, and we have low confident in the SSD representation, hence large weight is given to the document representation $\mathbf{d}_k$. On the contrary, $f(\theta_s)$ is close to 0.5 if θ_s provides a large degree of separation between the similar users and dissimilar users, then we have high confidence in the similar users and distribute equal weights to $\mathbf{d}_k$ and $\mathbf{d}_{ssd}$ for sentiment classification. Notice that gate vector $\mathbf{g}$ is automatically learned from the training data, while $g(\theta_s)$ encodes domain knowledge. The hybrid of data-driven and knowledge-driven gate values help to select the most informative features for classifying the document sentiments.

The proposed sentiment classifier then transforms $\mathbf{d}$ and $\mathbf{z}_{ij}$ into a predictive vector with a dimensionality that equals to the number of sentiment classes C.

$$\begin{aligned} \hat{\mathbf{y}} &= softmax(\mathbf{W}_d \mathbf{d} + \mathbf{W}_{upi} \mathbf{z}_{ij} + \mathbf{b}_y) \\ loss &= - \sum_{<u_i, p_j, d_k> \in \mathcal{D}} \sum_{c=1}^{C} y_c log(\hat{y}_c) \end{aligned} \tag{10}$$

where $\mathcal{D}$ is the training set, and $\mathbf{W}_d$, $\mathbf{W}_{upi}$ are weight matrices and $\mathbf{b}_y$ is a bias vector. $\hat{y}_c$ is the c-th element in predictive vector $\hat{\mathbf{y}}$, indicating the probability that the tuple $< u_i, p_j, d_k >$ is classified to class c. We define training objective function $loss$ as the cross-entropy error between predictive labels $\hat{\mathbf{y}}$ and the ground-truth labels $\{y_c\}_{c=1}^{C}$.

Experiment

In this section, we conduct experiments on publicly available datasets and present the corresponding results. To comprehensively evaluate the proposed method, we investigate the following research questions:

- **RQ1** Can the proposed method outperform the sate-of-the-art sentiment classification methods?
- **RQ2** Are the proposed UPI and SSD components helpful in improving the performance of sentiment analysis?
- **RQ3** How do the key hyperparameters affect the performance of the proposed model?
- **RQ4** What is the time complexity of the proposed method compared with the state-of-the-art methods?

Statistics	IMDB	Yelp 2013
#users	1310	1631
#products	1635	1635
#train docs	67426	62522
#dev docs	8381	7773
#test docs	9112	8671
#docs/user	64.82	48.42
#docs/product	51.94	48.36
#classes	10	5

Table 1: Statistics of the IMDB and Yelp 2013

Methods	IMDB		Yelp 2013	
	Acc.	RMSE	Acc.	RMSE
UPDMN	0.465	1.351	0.639	0.662
TUPCNN	0.488	1.451	0.639	0.694
NSC	0.533	1.281	0.650	0.692
HCSC	0.542	1.213	0.657	0.660
CSC	**0.578***	**1.198***	**0.679***	**0.646***

Table 2: Accuracy and RMSE of the proposed models and the baselines. * indicates that the improvements over all the other models are statistically significant for $p < 0.01$.

Datasets and Evaluation

We conduct experiments on two publicly accessible datasets: IMDB and Yelp 2013. These two dataset was collected by Tang et al. (2015) and divided into train, dev, and test sets with a 8:1:1 ratio. Textual reviews in the datasets are tokenized and sentence-splitted using Stanford CoreNLP (Manning et al. 2014). The statistics of the two datasets are present in Table.1.

Classification on the two datasets are measured using two metrics: accuracy and root-mean-square error (RMSE), where accuracy measures the overall performance of sentiment classification, and RMSE measures differences between predicted rating values and the ground-truth ones.

Implementation

We implement the proposed method with tensorflow [1] deep-learning library, and make the source codes publicly available to facilitate community research[2]. The dimensionality of word, user and product vectors are set to 300, and the word embedding matrix is initialized with pre-trained Glove embeddings (Pennington, Socher, and Manning 2014). We use stacked 1-dimension CNNs for extracting word representations. For the first layer CNN, the kernel sizes are set to 3 and 5, each with 128 feature maps. As for he second layer CNN, we use a kernel of size 5 with 300 feature maps. To prevent overfitting, we deploy dropout (Srivastava et al. 2014) layers upon all non-linear transformations with a dropout rate of 0.5. The implemented model is trained via stochastic gradient descent over shuffled mini-batches with a batch size of 128. The defined objective function is optimized using Adam (Kingma and Ba 2014) optimizer with an initial learning rate of 0.001. We perform early stopping and fine tune the parameters with the dev set. All of the experiments and training are done using a NVIDIA GeForce GTX 1070 graphics card with 8G memory.

Baselines

To demonstrate the effectiveness of the proposed method, it is compared with the following baselines:

- HCSC (Amplayo et al. 2018) models user/product representations from similar users/products to mitigate cold-start problem.
- NSC (Chen et al. 2016a) uses a hierarchical LSTM model and an attention mechanism for encoding documents. The user/product information is considered in the attention mechanism.
- UPDMN (Dou 2017) model document representations with a LSTM encoder and modifies the representations with other documents of the user/product as the memory.
- TUPCNN (Chen et al. 2016b) considers temporal rating behaviours for modeling temporal user/product embeddings, and extends the CNN-based classifier with the embeddings.

Among the baselines, HCSC achieves the state-of-the-art results. Notice that there are many other models for document-level sentiment classification, but they are demonstrated to be inferior to the aforementioned baselines in previous works (Amplayo et al. 2018; Chen et al. 2016a). Therefore, we do not compare with those models to avoid redundancy. Additionally, we also implement three versions of the proposed model to investigate the contributions of each of its components, i.e. (a) the CSC-BASE model that excludes both SSD and UPI components, (b) the CSC-UPI model that includes UPI component on the basis of CSC-BASE, and (c) CSC-SSD model that includes SSD component on the basis of CSC-BASE.

Performance Comparison (RQ1)

Table.2 shows the accuracy and RMSE on both datasets. We have the following observations:

- The proposed model, CSC, achieves the best performance in general, and outperform the base best baseline (i.e. HCSC) by a large margin across different datasets and metrics. This justifies the effectiveness of leveraging informative document of the products for collaborative sentiment classification.
- All the other models yield better results than UPDMN. Even though UPDMN exploits other documents of the users/products for sentiment analysis. However, other documents of the users/products does not necessarily have the similar sentiment, and the inconsistent sentiment of different documents can jeopardise the model and negatively affect its performance.
- HCSC and NSC incorporate user/product embeddings in learning attentive document representations, and they achieve significant improvement compared to UPDMN

[1] https://github.com/tensorflow

[2] https://github.com/uqjwen/rectiment

Methods	IMDB		Yelp 2013	
	Acc.	RMSE	Acc.	RMSE
CSC-BASE	0.522	1.320	0.633	0.706
CSC-UPI	0.541	1.264	0.640	0.688
CSC-SSD	0.572	1.211	0.668	0.659
CSC	**0.578***	**1.198***	**0.679***	**0.646***

Table 3: Accuracy and RMSE of the proposed models and its variants. * indicates that the improvements over all the other variants are statistically significant for $p < 0.01$.

and TUPCNN. This indicates the effectiveness of incorporating user/product preferences for sentiment analysis.

- HCSC and NSC both employ attention mechanisms on word vectors to obtain document representations which are then used for sentiment classification. Therefore, these two model are limited by the information conveyed in a single document. On the contrary, CSC is able to leverage the informative documents of the products for collaborative sentiment analysis and improve performance.

Efficacy of CSC Components(RQ2)

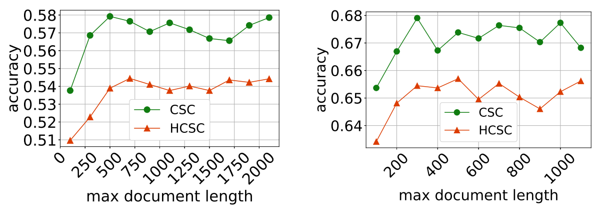
To investigate the effectiveness of each components (i.e. UPI and SSD) of CSC, we compare CSC with CSC-BASE, CSC-UPI and CSC-SSD in terms of accuracy and RMSE.

The comparison results are shown in Table.3, and from the table we have the following key observations:

- CSC-UPI yields better classification performance than CSC-BASE across different datasets and metrics, demonstrating the effectiveness of incorporating UPI for sentiment analysis. In CSC-UPI, the UPI component are included in the final sentiment prediction, which can learn-specific user/product embeddings. As the document representations are attentively calculated as a function of word, user and product embeddings, the task-specific user/product embeddings can help to learn better attentive document representations for sentiment analysis.
- The proposed model achieves better performance than its variants, CSC-UPI and CSC-SSD. The advantage over the variants justifies the efficacy of incorporating the components simultaneously. Specifically, speculation on the similar documents can capture informative documents for collaborative sentiment analysis, and modeling UPI can help to learn better document representations for improving performance.
- The improvement of CSC-SSD over CSC-BASE is more significant than that of CSC-UPI over CSC-BASE. This shows that SSD component are more effective in improving classification performance, as it is able to capture informative sentence for collaborative sentiment analysis.

Hyperparameter Study(RQ3)

Max document length To accelerate the training process, documents of various length are padded to a pre-defined maximum length. We investigate the impact of the maximum length on the model performance. The classification



(a) IMDB (b) Yelp 2013

Figure 2: Sentiment classification accuracy under different maximum document length.

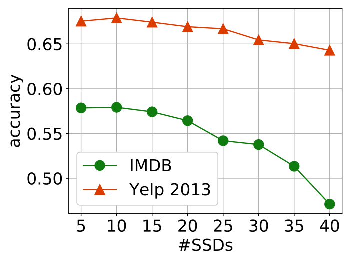


Figure 3: Sentiment classification accuracy under different number of specular similar documents.

performance is presented as a function of the maximum document length in Fig.2. From the figures, we can conclude that the proposed model outperforms the best baseline under different maximum document lengths, justifying the stability and robustness of CSC across different datasets. In Fig.2, the classification accuracy increases with maximum document length, and retains at a rather stable afterwards. This indicates that the first few hundreds words are sufficient enough to determine the document sentiments, and the additional word vectors do not significantly increase the classification performance. As the training time increase with the maximum document length, we experimentally select maximum document lengths to balance the tradeoff between accuracy and training overhead, specifically, 500 for IMDB and 300 for Yelp 2013.

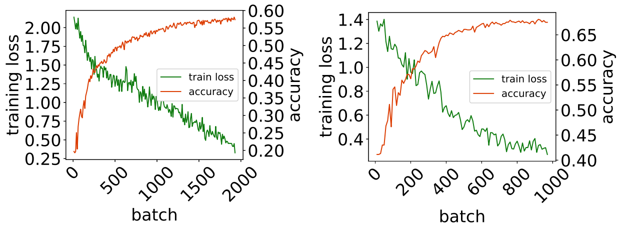
Number of SSDs (#SSDs) We study the impact that #SSDs has on the model performance, and present the result in Fig.3. From the figure we can see that the model performance remains at a stable level when #SSDs is smaller than 20, and the performance decreases dramatically as we increase #SSDs to 40. For products with fewer than 40 documents, we add repeatedly add current document as its SSDs until the predefined #SSDs is reached. The underlying reason is that more SSDs inevitably introduce noises, and affect the model performance. However, this experiment demonstrates that we can set #SSDs to wide range of choices to obtain the satisfactory performance.

Time Complexity Analysis(RQ4)

In the proposed model, the document representations are calculated by applying 1-dimension CNNs and an atten-

Models	IMDB	Yelp 2013
HCSC	69.538	43.095
CSC-BASE	13.013 (5.34x)	8.729 (4.94x)
CSC-UPI	13.398 (5.19x)	8.991 (4.79x)
CSC-SSD	13.317 (5.22x)	8.786 (4.91x)
CSC	13.524 (5.14x)	9.061 (4.76x)

Table 4: Training time (in seconds) for running 100 batches data of competing models across the datasets. The number in the parenthesis are the speedup of time compared to HCSC.



(a) IMDB (b) Yelp 2013

Figure 4: Training loss and validation accuracy in each batch on IMDB and Yelp 2013 .

tion mechanism on the word vectors. The convolution operations can be paralleled, making it time efficient to train the proposed model. We report the training time in seconds for running 100 batches of data, and present the results in Table.4. For fair comparison, the maximum document lengths in HCSC are set to be equal to those in CSC. As shown in the Table.4, HCSC takes 69 and 43 seconds to process 100 batches of data of IMDB and Yelp 2013 respectively, while CSC achieves 5.14x and 4.76x speedup on these two datasets. This is because HCSC uses Bidirectional Long Short Term Memory (BiLSTM) networks for extracting document representations, and they are non-parallelizable. Moreover, comparing CSC with its variants, the differences between their training overhead is marginal. This indicates that incorporating the SSD and UPI components yield better results without incurring much overhead.

Fig.4 presents the training loss and validation accuracy of the proposed model in each batch across the datasets. The proposed model only requires 4 and 2 epochs (i.e. 2108 and 972 batches) respectively to get the best result on IMDB and Yelp 2013. It justifies the necessity to perform early stopping to avoid overfitting.

Related Work

Sentiment classification is a fundamental task in the area of sentiment analysis (Pang, Lee, and Vaithyanathan 2002), and it can be tackled at different levels of granularity, from aspect level (Li et al. 2019a; Xu, Mao, and Chen 2019; Amplayo and Hwang 2017) to document level (Amplayo et al. 2018; Dou 2017) depends on the specific application. In this paper, we focus document-level sentiment classification, which recognizes the overall sentiment of a document that a user writes about a product. Recent studies (Diao et al. 2014; Yang et al. 2017) have demonstrated that incorporating additional contexts (e.g. user and product information) can significantly improve the classification performance. For example, in (Tang, Qin, and Liu 2015) utilize user/product text preference matrix to modify the semantic meaning of the words, and they apply convolution and average pooling operations to the words for modeling the semantic representations of sentences. Chen at al. (2016a) incorporate user product embeddings in modeling the document representations via an attention mechanism. HCSC (Amplayo et al. 2018) addresses the cold-start problem in sentiment classification. It leverages similar users/products to obtain additional shared document vectors. One main drawback of those works is that they are mainly based on local documents, and ignore other documents of the products that might be helpful for sentiment classification.

UPDMN (Dou 2017) is the state-of-the-art model that incorporate all the documents of the users/products for sentiment classification. However, it simply aggregate all the documents via an attention mechanism to comprise the predictive vector, which can inevitably introduce noises and negatively affect the model performance. On the contrary, The proposed model is based on the speculation that users with similar rating behaviours are more likely to write document of same sentiments toward a product, hence we are able to capture the informative documents for collaborative sentiment classification.

Conclusion

In this paper, we propose a novel collaborative sentiment classification model. It is based on the speculation that users with similar rating behaviours are more likely to write documents of the same sentiment toward a product, and we propose to exploit those informative documents for collaborative classification. We conduct experiments on two public datasets, and demonstrate the advantage of the proposed model over the state-of-the-art baselines. We also investigate different variants of the proposed model to reveal the rationale underlying the advantage of the proposed model. Finally, analyze the hyper-parameters and time complexity of the proposed model to investigate its robustness.

References

Amplayo, R. K., and Hwang, S.-w. 2017. Aspect sentiment model for micro reviews. In *2017 IEEE International Conference on Data Mining (ICDM)*, 727–732. IEEE.

Amplayo, R. K.; Kim, J.; Sung, S.; and won Hwang, S. 2018. Cold-start aware user and product attention for sentiment classification. In *Proceedings of 56th Annual Meeting of the Association for Computational Linguistics*, 2535–2544.

Chen, H.; Sun, M.; Tu, C.; Lin, Y.; and Liu, Z. 2016a. Neural sentiment classification with user and production attention. In *Proceedings of the 2016 Conference on Empirical Methods in Nutural Language Processing*, 1650–1659.

Chen, T.; Xu, R.; He, Y.; Xia, Y.; and Wang, X. 2016b. Learning user and product distributed representations using a sequence model for sentiment analysis. *IEEE Computational Intelligence Magazine* 34–44.

Diao, Q.; Qiu, M.; Chao-Yuan, W.; Smola, A. J.; Jiang, J.; and Wang, C. 2014. Jointly modelling aspects, ratngs and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 193–202.

Dou, Z.-Y. 2017. Capturing user and product information for document level sentiment analysis with deep memory network. In *Proceedings of the 2017 Conference on Empirical Methods in Nutural Language Processing*, 521–526.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee*, 173–182. ACM.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Nutural Language Processing*.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *Computer Science*.

Li, X.; Bing, L.; Li, P.; and Lam, W. 2019a. A unified model for opinion target extraction and target sentiment prediction. In *Proceedings of The Thirty-Third AAAI Conference on Artificial Intelligence*, 6714–6721.

Li, Z.; Wei, Y.; Zhang, Y.; Zhang, X.; and Li, X. 2019b. Exploiting coarse-to-fine task transfer for aspect-level sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4253–4260.

Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.; and McClosky, D. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 53th Annual Meeting of the Association for Computational Linguistics: system demonstrations*, 55–60.

McCann, B.; Bradbury, J.; Xiong, C.; and Socher, R. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, 6297–6308.

Ming, Z.; Li, F.; Sheng, W.; and Liu, S. 2014. Suit: A supervised user-item based topic model for sentiment analysis. In *Twenty-eighth AAAI Conference on Artificial Intelligence*.

Pang, B.; Lee, L.; and Vaithyanathan, S. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*, 1532–1543.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 1929–1958.

Taboada, M.; Brooke, J.; Tofiloski, M.; Voll, K.; and Stede, M. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics* 37(2):267–307.

Tai, K. S.; Socher, R.; and Manning, C. D. 2015. Improved semantic representations from tree-structured long short-term memory networks. *Computer Science* 5(1).

Tang, D.; Qin, B.; and Liu, T. 2015. Learning semantic representations of users and products for document level sentiment classificaiton. In *Proceedings of 53th Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 1014–1023.

Wang, X.; Lu, W.; Ester, M.; Wang, C.; and Chen, C. 2016. Social recommendation with strong and weak ties. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 5–14. ACM.

Wang, X.; Hoi, S. C.; Ester, M.; Bu, J.; and Chen, C. 2017. Learning personalized preference of strong and weak ties for social recommendation. In *Proceedings of the 26th International Conference on World Wide Web*, 1601–1610. International World Wide Web Conferences Steering Committee.

Xu, N.; Mao, W.; and Chen, G. 2019. Multi-interactive memory network for aspect based multimodal sentiment analysis. In *Proceedings of The Thirty-Third AAAI Conference on Artificial Intelligence*, 371–378.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 1480–1489.

Yang, M.; Mei, J.; Ji, H.; Zhao, Z.; Chen, X.; et al. 2017. Identifying and tracking sentiments and topics from social media texts during natural disasters. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 527–533.

Zheng, L.; Noroozi, V.; and Yu, P. S. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 425–434. ACM.